Fuzzy Optim. Model. J. 2026; 7(1) : 56-66

Fuzzy Optimization and Modeling Journal (FOMJ)

https://doi.org/10.57647/fomj.2026.0701.04

Original Research

An ABS-GA Algorithm for Solving Fuzzy Optimization Problems

Ali Mehrabian[1], Reza Ghanbari[1,*], Khatere Ghorbani-Moghadam[2]

[1]*Faculty of Mathematical Sciences, Department of Applied Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran*
[2]*Mosaheb Institute of Mathematics, Kharazmi University, Tehran, Iran*

*__Corresponding author:__ rghanbari@um.ac.ir

Article History

Received:
5 January 2026

Revised:
10 February 2026

Accepted:
27 March 2026

Published in Issue:
30 March 2026

Abstract:

This paper presents a fuzzy programming model with LR fuzzy coefficients. To solve it efficiently, we propose a novel hybrid ABS-GA algorithm that synergistically combines the ABS algorithm for dimensionality reduction with a Genetic Algorithm (GA). First, ABS projects the original n-dimensional problem into a reduced $(n-m)$-dimensional subspace using the linear constraints $Ax = b$, ensuring feasibility and shrinking the search space. Then, a tailored GA optimizes within this reduced space, employing Ghanbari et al. [1] $O(1)$ comparison formula for direct and efficient fuzzy number evaluation, and a novel tangent cone-based mutation operator for enhanced local exploration. Numerical experiments demonstrate that ABS-GA significantly outperforms existing methods in both solution quality and computational efficiency, validating the effectiveness of the integrated approach.

Keywords: Triangular intuitionistic fuzzy regression model (IFRM); Full IFRM; Triangular intuitionistic fuzzy numbers (TIFNs); Intuitionistic fuzzy least absolute of discrepancies (IFLAD); Homogeneity principleFuzzy optimization; Hybrid ABS-GA algorithm; Fuzzy comparison; LR fuzzy numbers; Genetic algorithm

Cite this article: Mehrabian A, Ghanbari R, Ghorbani-Moghadam Kh. An ABS-GA Algorithm for Solving Fuzzy Optimization Problems. Fuzzy Optim. Model. J. 2026;7(1): 56-66. https://doi.org/fomj.2026.0701.04

1. Introduction

Fuzzy sets serve as powerful mathematical tools for modeling collections of objects with imprecise or vague boundaries, finding significant applications in various real-world optimization problems under uncertainty. A fundamental challenge across many fuzzy methodologies, particularly in optimization, is the comparison of fuzzy numbers. Despite its necessity, there is no universally accepted method for this task.

Over the years, numerous approaches have been developed, drawing on ranking functions, α-cuts, fuzzy relations, and the extension principle, among others (see [1, 2, 3, 4, 5]). In fact, more than 40 distinct methods for comparing fuzzy numbers have been documented in the literature [6]. Generally, methods for comparing fuzzy numbers fall into two main categories. The first category comprises indirect methods (transformation-based approaches), such as ranking functions, which convert fuzzy numbers into crisp values for comparison. A major limitation of these methods is their lack of bijectivity, potentially leading to information loss during the transformation. The second category involves direct methods, which compare fuzzy numbers without prior conversion. While these methods preserve fuzzy information, they often entail substantial computational burdens.

Among the well-established direct methods, Kerre's method [7] stands out for ordering fuzzy numbers. This method traditionally involves a two-step process: first, computing the fuzzy maximum of two numbers via the extension principle or α-cuts, and then performing a comparison based on the Hamming distance [5, 6, 8]. While foundational, this process can be computationally intensive. Ghanbari et al. [9] first presented a result that led to a direct and efficient formula for computing the maximum of two arbitrary LR fuzzy numbers. Then, by applying the direct formula for $\widetilde{\max}$, they modified Kerre's method to compare two LR fuzzy numbers.

Finally, using their modified Kerre's method for comparing two LR fuzzy numbers, they established some simple formulas for the comparison of fuzzy triangular numbers. These formulas execute in constant time, $O(1)$, requiring only a few basic arithmetic operations and eliminating all numerical integration. The formulas that represented by Ghanbari et al. [9], have a major leap in efficiency for the triangular case. Despite this breakthrough, a critical limitation persists. The elegant and efficient $O(1)$ closed-form solution is exclusive to triangular fuzzy numbers. For the broader, more expressive, and often more realistic class of general LR fuzzy numbers which includes Gaussian, bell-shaped, and other non-linear types the framework by Ghanbari et al. [9], still reverts to evaluating definite integrals within the comparison formulas. Furthermore, due to the important applications of bell-shaped fuzzy numbers in various fields, and considering the computational challenges associated with direct comparison methods, previous studies have turned to metaheuristic algorithms to handle complex fuzzy optimization problems. The relevance of these methods is underscored by the wide application of LR fuzzy numbers in domains requiring the handling of uncertainty. For instance, in medical diagnosis, they help manage imprecise patient data, leading to more accurate clinical decisions [10, 11]. Their utility extends to the development of bell-shaped fuzzy numbers for statistical modeling and AI systems [12], and even to specialized areas like fingerprint authentication, where they enhance the accuracy and efficiency of matching algorithms [13].

The ability to compare fuzzy numbers is crucial in fuzzy optimization. A common theme in many solution approaches is the transformation of a fuzzy problem into an equivalent crisp one [14]. Alternatively, methods that work within the fuzzy domain itself have been proposed. Bellman and Zadeh [15] introduced a seminal symmetric model where goals and constraints are treated as fuzzy sets, and the optimal decision is defined as the intersection of these sets, solved by a max-min approach:

$$\max \min_{k} \mu_k(x) \quad k = 1, \ldots, m,$$

where $\mu_k(x)$ is the membership function for the k-th goal or constraint. Ribeiro and Pires [16] applied the simulated annealing algorithm to solve this max-min model, highlighting the need for randomized transitions and a carefully defined annealing schedule [17, 18]. In continuation of this methodological approach, GAs emerge as a natural choice for implementing our fuzzy optimization framework, owing to their flexibility and robust performance in handling uncertain parameters. GAs have also been effectively employed in fuzzy optimization. Lin [19] used a GA to solve linear programming problems with fuzzy constraints, where the symbol '$\preceq$' allows for minor, acceptability-weighted violations. Other notable contributions include regression models with triangular interval-valued fuzzy numbers [20, 21], Pythagorean fuzzy linear programming for multi-criteria decision-making [22], and methods for linear programming with interval-valued intuitionistic fuzzy parameters [23]. For a comprehensive survey of models and methods for fuzzy linear programming, see [24]. One can see other works in [25, 26, 27, 28, 29, 30, 31].

Despite significant advancements in fuzzy optimization over the past decades, several critical research gaps remain unaddressed. First, while numerous methods have been developed for comparing fuzzy numbers, they generally fall into two categories with inherent limitations: indirect methods (e.g., ranking functions) suffer from information loss during conversion to crisp values, while direct methods (e.g., Kerre's method) often entail substantial computational burdens due to numerical integration requirements. Second, although Ghanbari et al. [9] made a significant breakthrough by introducing an $O(1)$ comparison formula for triangular fuzzy numbers, their efficient closed-form solution is exclusive to the triangular case. For the broader and more realistic class of general LR fuzzy numbers which includes Gaussian, bell-shaped, and other non-linear types commonly encountered in machine learning, medical diagnosis, and statistical modeling their framework still reverts to evaluating definite integrals, creating a computational bottleneck. Third, existing metaheuristic approaches for fuzzy optimization, such as standard Genetic Algorithms or Simulated Annealing, typically handle constraints through penalty functions or repair mechanisms. These approaches become increasingly inefficient as problem dimensionality grows, as they must expend substantial computational effort exploring infeasible regions. Furthermore, they often struggle to maintain diversity and convergence when the feasible region is defined by a large number of linear equality constraints. Fourth, to the best of our knowledge, no prior work has exploited the synergy between exact algebraic dimensionality reduction techniques (such as the ABS algorithm) and evolutionary algorithms for solving fuzzy optimization problems. This gap is particularly significant given that many real-world fuzzy linear programming problems involve large numbers of equality constraints, where dimensionality reduction could yield substantial computational benefits.

Following this methodological rationale, we adopt a hybrid ABS-GA approach that synergistically integrates the ABS[1] algorithm for dimensionality reduction with a Genetic Algorithm (GA) for global optimization. The ABS algorithm plays a pivotal role in our framework. By leveraging the linear equality constraints $Ax = b$ of our fuzzy linear programming model, ABS analytically projects the original n-dimensional problem into a reduced $(n-m)$-dimensional subspace. This exact dimensionality reduction not only drastically shrinks the search space but also ensures that all candidate solutions automatically satisfy the linear constraints, thereby eliminating the need for constraint-handling mechanisms within the GA and allowing it to focus exclusively on

[1] Abaffy-Broyden-Spedicato

optimizing the objective function. Once the problem is transformed into this lower-dimensional form, a customized GA operates within the reduced search space. Crucially, to perform fitness evaluations and comparisons between fuzzy objective values, we leverage the efficient $O(1)$ comparison formula for LR fuzzy numbers introduced by Ghanbari et al. [1]. This allows for direct and computationally cheap comparisons without converting fuzzy numbers to crisp values, thereby preserving fuzzy information while avoiding numerical integration costs. To further enhance the search capability of the GA in this reduced space, we introduce a novel mutation operator based on an approximation of the tangent cone. This operator intelligently explores the neighborhood of a solution by generating feasible directions derived from the polar cone of ε-active constraints, effectively balancing exploration and exploitation. Thus, the proposed ABS-GA algorithm combines the structural preprocessing of ABS with the evolutionary power of GA and the computational efficiency of Ghanbari et al. [1] comparison method, resulting in a robust and efficient solver for fuzzy optimization problems.

In this paper, we present a fuzzy optimization model solved by a novel hybrid algorithm. The main contributions of this work are threefold. First, we propose the ABS-GA algorithm, which synergistically combines the ABS algorithm for exact dimensionality reduction with a Genetic Algorithm (GA) for global search. Unlike conventional methods that handle constraints via penalty functions or repair mechanisms, our approach analytically projects the original problem onto the null space of the constraint matrix, eliminating equality constraints and dramatically reducing the search space while guaranteeing feasibility. Second, we introduce a novel mutation operator based on an approximation of the tangent cone, which enables intelligent exploration of the feasible region by generating directions derived from the polar cone of ε-active constraints. This operator effectively balances exploration and exploitation, enhancing the GA's ability to locate high-quality solutions. Third, for all fuzzy number comparisons required by the algorithm, we leverage the recently developed $O(1)$ comparison formula for general LR fuzzy numbers introduced by Ghanbari et al. [1], which allows for direct comparisons without conversion to crisp values, preserving fuzzy information while avoiding numerical integration costs. To the best of our knowledge, this is the first work that integrates ABS preprocessing with a tangent cone-based GA for solving fuzzy linear programming problems with LR fuzzy coefficients, offering a computationally efficient and theoretically sound alternative to existing approaches.

The rest of this paper is organized as follows: section 2 provides the necessary preliminaries on fuzzy ordering and LR fuzzy numbers. Our proposed hybrid ABS-GA algorithm is detailed in section 3. Numerical results and a performance analysis are presented in section 4. Finally, we conclude in section 5.

2. Preliminaries

This section reviews the fundamental definitions and the specific comparison method that underpin our work.

2.1 Fuzzy numbers and LR-type

We begin with basic definitions related to fuzzy numbers.

Definition 1. [32] A fuzzy number is a fuzzy quantity A satisfying the following conditions:

1. $\mu_A(x) = 1$ for exactly one x.
2. The support $\{x : \mu_A(x) > 0\}$ of A is bounded.
3. Every α-cut, $\{x : \mu_k(x) \geq \alpha\}$, of A is a closed interval.

Definition 2. [11] A decreasing function $L : R^+ \to [0,1]$ is called a shape function if it satisfies:

$$\begin{cases} L(0) = 1, \\ L(1) = 0, \\ 0 < L(x) < 1, \quad for\, x \neq 0,1. \end{cases}$$

Definition 3. [11] A fuzzy number $\tilde{A}$ is of **LR-type** if there exist shape functions L (for left), R (for right), and scalars $\alpha > 0$, $\beta > 0$ such that:

$$\mu_{\tilde{A}}(x) = \begin{cases} L(\frac{a-x}{\alpha}), & x \leq a \\ R(\frac{x-a}{\beta}), & x \geq a \end{cases}$$

The mean value of $\tilde{A}$ is a, and α, β are the left and right spreads, respectively. Symbolically, $\tilde{A}$ is denoted by $(a-\alpha/a/a+\beta)_{\text{LR}}$.

Remark 1. Based on **Definition 3**, an LR fuzzy number $\tilde{A}$ can also be represented as $\tilde{A} = (A_L, A_R)$, where A_L is the shape function for the left arm and A_R is the shape function for the right arm.

2.2 Kerre's method to compare two fuzzy numbers

In [6], the fuzzy max ($\widetilde{\max}$)of two fuzzy number (based on the extension principle) was introduced as follows:

$$\widetilde{\max}(\tilde{M},\tilde{N})(z) = \sup\{\min\left(\tilde{M}(x),\tilde{N}(y)\right) \mid \max(x,y) = z\}. \quad (1)$$

Remark 2. The reader can find a detailed study of properties of $\widetilde{\max}$ in [4].

Definition 4. Based on Kerre's method, we say $\tilde{M} \leq \tilde{N}$ if and only if

$$d(\tilde{N},\widetilde{\max}(\tilde{M},\tilde{N}) \leq d(\tilde{M},\widetilde{\max}(\tilde{M},\tilde{N}))$$

where $d(\cdot,\cdot)$ is the Hamming distance as defined in [6]. Similarly, "$\geq$" and "$=$" are defined.

Definition 5. Suppose $\tilde{M} = (a/b/c)_{LR}$ and $\tilde{N} = (a'/b'/c')_{LR}$ are two arbitrary LR fuzzy numbers, and let $\tilde{O} = \max(\tilde{M},\tilde{N})_{\text{LR}}$. Define

$$r(\tilde{M},\tilde{N}) = d(\tilde{M},\tilde{O}) - d(\tilde{N},\tilde{O})$$

In the Kerre's method [6],

- if $d(\tilde{M},\tilde{O}) - d(\tilde{N},\tilde{O}) \geq 0$, then $\tilde{M} \leq \tilde{N}$
- if $d(\tilde{M},\tilde{O}) - d(\tilde{N},\tilde{O}) \leq 0$, then $\tilde{M} \geq \tilde{N}$

https://doi.org/10.57647/fomj.2026.0701.04

2.3 Ghanbari et al.'s comparison formula for LR fuzzy numbers

Ghanbari et al. [9] introduced an efficient comparison formula that operates in $O(1)$ time. However, the formula presented in their work was specifically designed for triangular fuzzy numbers, and they subsequently solved a fuzzy optimization problem with triangular numbers using a metaheuristic algorithm. In contrast, our work is motivated by the widespread applications in machine learning and the prevalence of normal distribution curves in real-world problems. Consequently, we employ quadratic LR fuzzy numbers in our framework. Subsequently, Ghanbari et al. have established generalized formulas for LR fuzzy number comparison in [1], which form the basis of our comparison methodology. So, for comparing fuzzy numbers within our optimization algorithm, we use the formula that is introduced by Ghanbari et al. [1].

Theorem 1. [1] Let $\tilde{M} = (a/b/c)_{\mathrm{LR}}$ and $\tilde{N} = (a^{'}/b^{'}/c^{'})_{\mathrm{LR}}$ as two LR fuzzy numbers, by using simpson's approximation method we have six cases as follows:

(In this theorem, we define $\min(a,a^{'}) = \underline{a}$, $\max(a,a^{'}) = \overline{a}$, $\min(c,c^{'}) = \underline{c}$, and $\max(c,c^{'}) = \overline{c}$ and also $\overline{x}$ is the length of the intersection point of M_R and N_L).

1. If $c \leq a^{'}$ then we have:

$$r(\tilde{M},\tilde{N}) = \frac{2}{3}\left[(b-a)M_L\left(\frac{a+b}{2}\right) + (c-b)M_R\left(\frac{b+c}{2}\right) + (b^{'}-a^{'})N_L\left(\frac{a^{'}+b^{'}}{2}\right) + (c^{'}-b^{'})N_R\left(\frac{b^{'}+c^{'}}{2}\right)\right] + \frac{1}{6}[c-a+c^{'}-a^{'}]. \tag{2}$$

2. If $b = b^{'}$ then we have:

$$\begin{aligned} r(\tilde{M},\tilde{N}) = {} & \frac{\overline{a}-\underline{a}}{6}\Bigg[M_L(\underline{a}) - N_L(\underline{a}) + 4\left(M_L(\frac{\overline{a}+\underline{a}}{2}) - N_L(\frac{\overline{a}+\underline{a}}{2})\right) + M_L(\overline{a}) - N_L(\overline{a})\Bigg] + \\ & \frac{b-\overline{a}}{6}\Bigg[M_L(\overline{a}) - N_L(\overline{a}) + 4\left(M_L(\frac{\overline{a}+b}{2}) - N_L(\frac{\overline{a}+b}{2})\right)\Bigg] + \\ & \frac{c-b}{6}\Bigg[4\left(N_R(\frac{b+\underline{c}}{2}) - M_R(\frac{b+\underline{c}}{2})\right) + N_R(\underline{c}) - M_R(\underline{c})\Bigg] + \\ & (\frac{\overline{c}-\underline{c}}{6})[N_R(\underline{c}) - M_R(\underline{c}) + 4\left(N_R(\frac{\overline{c}+c}{2}) - M_R(\frac{\overline{c}+c}{2})\right) + N_R(\overline{c}) - M_R(\overline{c})]. \end{aligned} \tag{3}$$

3. If $(b < b^{'})$, $(b^{'} \leq c)$ and $(a^{'} \leq b)$ then we have:

$$\begin{aligned} r(\tilde{M},\tilde{N}) = {} & \frac{\overline{a}-\underline{a}}{6}\Bigg[M_L(\underline{a}) - N_L(\underline{a}) + 4\left(M_L(\frac{\overline{a}+\underline{a}}{2}) - N_L(\frac{\overline{a}+\underline{a}}{2})\right) + M_L(\overline{a}) - N_L(\overline{a})\Bigg] + \\ & \frac{b-\overline{a}}{6}\Bigg[M_L(\overline{a}) - N_L(\overline{a}) + 4\left(M_L(\frac{\overline{a}+b}{2}) - N_L(\frac{\overline{a}+b}{2})\right) + 1 - N_L(b)\Bigg] + \\ & \frac{\overline{x}-b}{6}\Bigg[1 - N_L(b) + 4\left(M_R(\frac{b+\overline{x}}{2}) - N_L(\frac{b+\overline{x}}{2})\right) + M_R(\overline{x}) - N_L(\overline{x})\Bigg] + \\ & \frac{b^{'}-\overline{x}}{6}\Bigg[N_L(\overline{x}) - M_R(\overline{x}) + 4\left(N_L(\frac{\overline{x}+b^{'}}{2}) - M_R(\frac{\overline{x}+b^{'}}{2})\right) + 1 - M_R(b^{'})\Bigg] + \\ & \frac{c-b^{'}}{6}\Bigg[1 - M_R(b^{'}) + 4\left(N_R(\frac{b^{'}+\underline{c}}{2}) - M_R(\frac{b^{'}+\underline{c}}{2})\right) + N_R(\underline{c}) - M_R(\underline{c})\Bigg] + \\ & \frac{\overline{c}-\underline{c}}{6}\Bigg[N_R(\underline{c}) - M_R(\underline{c}) + 4\left(N_R(\frac{\overline{c}+\underline{c}}{2}) - M_R(\frac{\overline{c}+\underline{c}}{2})\right) + N_R(\overline{c}) - M_R(\overline{c})\Bigg] \end{aligned} \tag{4}$$

4. If $b < b^{'}$, $a^{'} \leq b$ and $c \leq b^{'}$ then we have:

$$\begin{aligned} r(\tilde{M},\tilde{N}) = {} & \frac{\overline{a}-\underline{a}}{6}\Bigg[M_L(\underline{a}) - N_L(\underline{a}) + 4\left(M_L(\frac{\overline{a}+\underline{a}}{2}) - N_L(\frac{\overline{a}+\underline{a}}{2})\right) + M_L(\overline{a}) - N_L(\overline{a})\Bigg] + \\ & \frac{b-\overline{a}}{6}\Bigg[M_L(\overline{a}) - N_L(\overline{a}) + 4\left(M_L(\frac{\overline{a}+b}{2}) - N_L(\frac{\overline{a}+b}{2})\right) + 1 - N_L(b)\Bigg] + \\ & \frac{\overline{x}-b}{6}\Bigg[1 - N_L(b) + 4\left(M_R(\frac{b+\overline{x}}{2}) - N_L(\frac{b+\overline{x}}{2})\right) + M_R(\overline{x}) - N_L(\overline{x})\Bigg] + \\ & \frac{c-\overline{x}}{6}\Bigg[N_L(\overline{x}) - M_R(\overline{x}) + 4\left(N_L(\frac{\overline{x}+c}{2}) - M_R(\frac{\overline{x}+c}{2}) + N_L(c)\right)\Bigg] + \\ & \frac{b^{'}-c}{2}\Bigg[N_L(c) - M_R(\overline{x}) + 4\left(N_L(\frac{c+b^{'}}{2}) + 1\right)\Bigg] + \\ & \frac{c^{'}-b^{'}}{6}\left[1 + 4\left(N_R(\frac{b^{'}+c^{'}}{2})\right)\right]. \end{aligned} \tag{5}$$

https://doi.org/10.57647/fomj.2026.0701.04

5. If $b < b^{'}$, $b^{'} \leq c$ and $b \leq a^{'}$ then we have:

$$r(\tilde{M},\tilde{N}) = \frac{b-a}{6}\left[4M_L(\frac{a+b}{2})+1\right]+ \frac{a^{'}-b}{6}\left[1+4\left(M_R(\frac{b+a^{'}}{2})-M_R(a^{'})\right)\right]+ \frac{\bar{x}-a^{'}}{6}\left[M_R(a^{'})+4\left(M_R(\frac{a^{'}+\bar{x}}{2})-N_L(\frac{a^{'}+\bar{x}}{2})\right)+ M_R(\bar{x})-N_L(\bar{x})\right]+\frac{b^{'}-\bar{x}}{6} \left[N_L(\bar{x})-M_R(\bar{x})+4\left(N_L(\frac{\bar{x}+b^{'}}{2})-M_R(\frac{\bar{x}+b^{'}}{2})\right)+ 1-M_R(b^{'})\right]+\frac{\underline{c}-b^{'}}{6}\left[1-M_R(b^{'})+ 4\left(N_R(\frac{b^{'}+\underline{c}}{2})-M_R(\frac{b^{'}+\underline{c}}{2})\right)+N_R(\underline{c})-M_R(\underline{c})\right]+ \frac{\bar{c}-\underline{c}}{6}\left[N_R(\underline{c})-M_R(\underline{c})+ 4\left(N_R(\frac{\underline{c}+\bar{c}}{2})-M_R(\frac{\underline{c}+\bar{c}}{2})\right)+ N_R(\bar{c}-M_R(\bar{c}))\right]. \tag{6}$$

6. if $b < b^{'}$, $c \leq b^{'}$ and $b \leq a^{'}$ then we have:

$$r(\tilde{M},\tilde{N}) = \frac{b-a}{6}\left[4M_L(\frac{a+b}{2})+1\right]+ \frac{a^{'}-b}{6}\left[1+4\left(M_R(\frac{b+a^{'}}{2})+M_R(a^{'}\right)\right]+ \frac{\bar{x}-a^{'}}{6}\left[M_R(a^{'})+4\left(M_R(\frac{a^{'}+\bar{x}}{2})-N_L(\frac{a^{'}+\bar{x}}{2})\right)+ M_R(\bar{x})-N_L(\bar{x})\right]+\frac{c-\bar{x}}{6}\left[N_L(\bar{x})-M_R(\bar{\bar{x}})+ 4\left(N_L(\frac{\bar{x}+c}{2})-M_R(\frac{\bar{x}+c}{2})\right)+N_L(c)\right)\right]+ \frac{b^{'}-c}{6}\left[N_L(c)+4\left(N_L(\frac{c+b^{'}}{2}\right)+1\right]+ \frac{c^{'}-b^{'}}{6}[1-M_R(\underline{c})+4(N_R(\frac{b^{'}+c^{'}}{2}))\Big] \tag{7}$$

The formulas presented in this theorem operate in $O(1)$ time complexity and preserve the original fuzzy structure without requiring transformation. This efficient comparison mechanism enables the solution of general fuzzy optimization problems of the form,

$$\begin{cases} \min & \tilde{f}(x) \\ s.t. & Ax = b, \quad x \geq 0. \end{cases} \tag{8}$$

However, due to the extensive applications of linear programming in real-world scenarios, our current focus is specifically on fuzzy linear programming models. The key distinction between these formulas and those originally proposed by Ghanbari et al. [9] lies in their scope: while Ghanbari et al.'s initial work provided comparison methods specifically for triangular fuzzy numbers, the formulas presented here are developed for general LR fuzzy numbers, making them applicable to a broader class of problems, particularly those involving quadratic LR fuzzy numbers commonly encountered in machine learning and normal distribution-based applications.

3. ABS-GA algorithm for solving fuzzy linear programming problems

Consider the following optimization problem:

$$\begin{cases} \min & \tilde{c}^T x, \\ s.t. & Ax = b, \quad x \geq 0. \end{cases} \tag{9}$$

where $\tilde{c}_j$ for $j = 1, \dots, n$ are LR fuzzy coefficients of the objective function, $A \in \mathbb{R}^{m \times n}$ is a full row-rank matrix ($rank(A) = m$), and $x \in \mathbb{R}^n$ is the decision variable.

This fuzzy linear programming model has numerous practical applications in real world decision-making under uncertainty. As examples, in supply chain management, it can optimize logistics networks with uncertain transportation costs and demand fluctuations [33, 34]. In portfolio optimization, the model handles imprecise expected returns and risk parameters in financial investment decisions [35]. Additionally, in energy planning, it facilitates optimal power generation and distribution under uncertain demand and renewable energy output [36]. Given that the objective function involves fuzzy numbers while the constraints remain crisp, conventional penalty function methods cannot be directly applied, as the constraints cannot be incorporated into the objective function. Consider the system $Ax = b$, where $A = (a_1^T, a_2^2, \dots, a_m^T)$ and $b = (b_1, b_2, \dots, b_m)^T$. The general solution to this system is given by

$$x = x_p + N(A)q,$$

where x_p is a particular solution to the system of equations, and $N(A)$ represents the null space of the matrix A. The ABS method, first introduced by Abaffy, Broyden, and Spedicato in 1984 [1], performs this task iteratively. In their seminal work, they addressed the solution of linear algebraic equations through what became known as the unscaled or basic ABS class. Assume that $b^{(i)} = (b_1, \dots, b_i)^T$, that $A_i = (a_1, \dots, a_i)$, and that x_i is a solution to the system of equations with the first $i-1$ equations. Let H_i^T denote the null space of the matrix A_{i-1}^T. Given these assumptions, The ABS algorithm is described in Algorithm 1.

After running the algorithm, x_{m+1} will be a solution to the system of equations and H_{m+1}^T will be the null space of matrix A. Therefore, the general solution to the system is as follows:

$$y = H_{m+1}^T q + x_{m+1}$$

To address this, we employ the ABS algorithm (Algorithm 1), which transforms Problem (9) into the equivalent formulation (10). This transformation reduces the

Algorithm 1. ABS Algorithm.

Inputs: A, b
Outputs: x_{m+1}, H_{m+1}
Initialize: Choose arbitrary $x^1 \in \mathbb{R}^n$ and nonsingular $H^1 \in \mathbb{R}^{n\times n}$.
for $i = 1,2,\dots,m$ do:
    Compute residual: $t_i = a_i^T x_i - b_i$.
    Compute projection: $s_i = H_i a_i$.
    if $s_i = 0$ and $t_i = 0$ then
        Set $x_i^{+1} = x_i, H_i^{+1} = H_i$ {Redundant constraint}.
    else if $s_i = 0$ and $t_i \neq 0$ then
        Stop: System is incompatible.
    else if $s_i \neq 0$ then
        Choose z_i such that $z_i^T s_i \neq 0$.
        Compute search direction: $p_i = H_i^T z_i$.
        Compute step size: $\alpha_i = \frac{t_i}{(a_i^T p_i)}$
        Update solution: $x_{i+1} = x_i - \alpha_i p_i$.
        Choose w_i such that $w_i^T s \neq 0$.
        Update projection matrix $H_{i+1} = \frac{H_i a_i w_i^T H_i}{w_i^T H_i a_i}$.
    end if
end for

problem dimension by m units, leading to the following reduced-dimensional problem in $q \in \mathbb{R}^{n-m}$

$$\begin{cases} \min & \tilde{c}^T (H^t q + v) \\ s.t. & H^T q + v \geq 0 \end{cases} \tag{10}$$

The convex combination of multiple points lies within their convex hull, a property we leverage in the crossover operator to enhance exploitation. To maintain exploration capability, we employ cones in both the mutation operator and the initialization phase, enabling the algorithm to probe diverse regions of the solution space.

3.1 Cone and polar cone

A set K is called a cone if for every $x \in K$ and every $\alpha \geq 0$, we have $\alpha x \in K$. For any cone K, its polar cone is defined as:

$$K^\circ = \{y \in \mathbb{R}^n : \langle y, x\rangle \leq 0 \; for\; all\; x \in K\},$$

where $\langle y, x\rangle$ denotes the inner product of vectors y and x. Geometrically, the polar cone consists of all vectors that form an angle of at least 90° with every vector in K, as illustrated in figure 1.

Let $K \subseteq \mathbb{R}^n$ be a cone. A set $\varsigma = \{d^{(1)}, d^{(2)}, \dots, d^{(p)}\}$ is called a generator of K if

$$K = \{\sum_{i=1}^{p} \lambda_i d^{(i)} | \lambda_i \geq 0 \; for\; i = 1,2,\dots,p\}.$$

In other words, ς generates K if K consists of all nonnegative linear combinations of the vectors in ς.

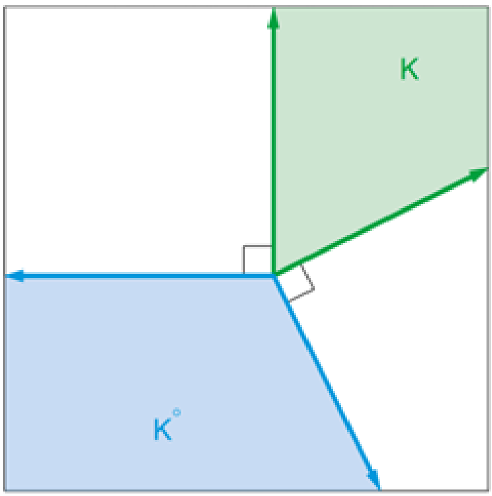

Figure 1. A cone K and its polar cone K° [37].

Now, consider a polyhedron $\omega = \{x \in \mathbb{R}^n : Ax \leq b\}$, where a_i^T denotes the ith row of A. For a point $x \in \omega$ and a tolerance $\varepsilon > 0$, define the index set of ε-active constraints as:

$$I(x,\varepsilon) = \{i \left| a_i^T x - b_i \right| < \varepsilon\}.$$

For each $i \in I(x,\varepsilon)$, define the vector $v_i(x,\varepsilon) = a_i$. Let $k(x,\varepsilon)$ be the cone generated by the vectors $v_i(x,\varepsilon)$ for $i \in I(x,\varepsilon)$. The significance of $k(x,\varepsilon)$ lies in the fact that, for an appropriate choice of ε, the translated polar cone $x + k^\circ(x,\varepsilon)$ provides a local approximation of the feasible region around x. In general, determining the generators of $k^\circ(x,\varepsilon)$ from those of $k(x,\varepsilon)$ is a combinatorial challenge equivalent to enumerating the vertices of a polyhedron. However, this complexity is significantly reduced if the generators of $k(x,\varepsilon)$ are linearly independent, in which case established methods can be applied [38].

Theorem 2. [38] For a given $\varepsilon > 0$, suppose the cone $k(x,\varepsilon)$ has a linearly independent generator matrix V. Let N be a matrix whose columns form a positive basis for the null space of V^T. Then the polar cone $k^\circ(x,\varepsilon)$ is generated by the columns of the matrix:

$$[N - V(V^TV)^{-1}\ V(V^TV)^{-1}].$$

3.2 Genetic algorithm implementation:

We employ a GA to solve the reduced-dimensional optimization problem (10). The algorithm begins by generating an initial population where all chromosomes satisfy the constraint $H^Tq+v \geq 0$. The implementation details of Algorithm 2 and Algorithm 3 are elaborated below.

Let $\Omega = \{q|H^Tq+v \geq 0\}$ be the feasible region, and let $x \in \Omega$ be an initial feasible point. Our objective is to generate additional feasible points within Ω. As previously discussed, for a suitably chosen ε, the cone $k(x,\varepsilon)$ captures the active constraints at x, while its polar cone $k^\circ(x,\varepsilon)$ provides directions that approximate the feasible region. Applying QR decomposition to $T = V(V^TV)^{-1}$ yields $TR^T = Q$, and a positive basis for the null space of V^TT is given by $\pm(I - TV^T)$. These basis directions, which are parallel to the boundary of the feasible set, serve as effective exploration directions. For a given direction d and the constraints h_i for $i = 1,2,\ldots,p$, the maximum step length α that maintains feasibility is computed as:

$$\alpha = \min_{i:h_i^Td<0}\left\{\frac{-(v_i+h_i^Tx)}{h_i^Td}\right\} \tag{11}$$

The actual step length is taken as $\min\{\alpha,1\}$. This procedure allows us to generate new feasible points from an existing point by moving along all exploration directions. The process is repeated iteratively from newly generated points until the desired initial population size is achieved.

The first feasible point in Ω can be obtained using linear programming solvers. Since Ω is convex, any convex combination of feasible points remains within Ω. This property is exploited in the crossover operator to produce new offspring. For mutation, we employ the same directional exploration method used in the initial population generation. After creating offspring and mutants, they are merged with the current population. The combined population is then sorted using Ghanbari et al. comparison formula for LR fuzzy numbers, and the top individuals are selected to form the new generation.

Due to the properties of scalar multiplication for trapezoidal fuzzy numbers, the objective function expression $\tilde{c}^T(H^Tq+v)$ cannot be algebraically simplified. Therefore, each candidate solution must be evaluated by direct computation according to the original problem formulation.

4. Numerical results

To comprehensively evaluate the performance of the proposed ABS-GA algorithm, we conducted extensive numerical experiments on a diverse set of fuzzy linear programming problems. This section details the experimental setup, performance metrics, comparative analysis, and statistical validation of the results. To generate problem instances in the form of (9), we employ the following procedure. First, we generate three random numbers in the interval $[-50,100]$ and sort them to define the parameters of an LR fuzzy number. This procedure is designed to produce quadratic LR fuzzy numbers. The LR fuzzy numbers generated by our procedure are quadratic LR fuzzy numbers, characterized by quadratic left and right shape functions. Subsequently, we generate an $m \times n$ constraint coefficient matrix with entries uniformly distributed in $[0,100]$. To ensure feasibility of the generated problem, we create a random vector $x \geq 0$ within the range $[0,100]$ and compute $b = Ax$. The experiments were conducted on a system with an Intel(R) Core(TM) i7-10700K CPU @ 3.8 GHz, 32GB RAM,

Algorithm 2. Initial population generation.

Inputs: H, v
Outputs: Feasible solutions
1: Find a feasible point x in the feasible region Ω and initialize population: pop $\leftarrow \{x\}$.
2: **while** $|pop| < desired\_size$ **do**
3: **for** each direction d in $[\pm T \pm (I - TV^T)]$ (for point x) **do**
4: Compute step length α for direction d and point x using (11).
5: Generate new point: $x' \leftarrow x + \alpha d$.
6: Update population: $pop \leftarrow pop \cup \{x'\}$.
7: **end for**
8: Select a new point x from the current population pop.
9: **end while**

https://doi.org/10.57647/fomj.2026.0701.04

Algorithm 3. ABS-GA hybrid algorithm.

Inputs: N: Population size (number of individuals), T: Maximum number of generations,
N_c: Number of offspring to generate per generation, N_m: Number of mutants to generate per generation.
Outputs: best chromosome in population.
1: **Dimensionality reduction:** Execute the ABS algorithm (Algorithm 1) to reduce the problem's search space.
2: **Population initialization:** Generate the initial population of candidate solutions using Algorithm 2
3: **Main optimization loop:**
for $i = 1$ to T **do:**
3.1. Offspring generation:
  for each j in $1, \ldots, N_c$ **do:**
    Create the j-th offspring via crossover and selection mechanisms.
  end for
3.2. Mutation phase:
  for each k in $1, \ldots, N_m$ **do**
    Generate the k-th mutant using the novel tangent cone-based mutation operator.
  end for
3.3. Population update:
    Combine the current population, offspring, and mutants into an extended candidate pool.
    Sort the combined pool based on fitness values (using Ghanbari et al.'s formulas for fuzzy numbers).
    Retain the top N individuals to maintain the fixed population size.
13: **end for**

running Windows 11 Pro, and the code was implemented in MATLAB 2021.

Having generated the problem instances, we can now solve them using Algorithm 3 (ABS-GA). An alternative approach for solving the problem is to apply the GA [39] directly without using the ABS algorithm for dimension reduction. In this case, we deal with two types of constraints: $Ax = b$ and $x \geq 0$ The termination criterion for this direct GA is set such that its maximum runtime does not exceed 20% more than the runtime of the ABS-GA algorithm. Based on empirical experimentation, the parameters are set as follows: population size of 80, number of algorithm iterations set to 90, number of parents for generating offspring via convex combination set to 4, and number of mutants per iteration set to 15.

As evident from the generated table displaying r (Algorithm 3, timed GA), the solutions obtained by Algorithm 3 consistently outperform those of the Timed GA across all test instances. This superiority can be attributed to the application of the ABS algorithm, which effectively reduces the problem dimensionality. The problem dimensions were intentionally designed with m and n being close in value, thereby enabling the ABS algorithm to achieve a substantial reduction in the problem size and consequently enhance computational efficiency. In Table 1, the column entitled Problem size shows the dimensions of the test problems, the column entitled r (Algorithm 3, timed GA) displays the performance ratio between two approaches: the objective function value obtained by the standard GA versus that obtained by Algorithm 3 (our proposed ABS-GA method). It is important to note that in Algorithm 3 and Algorithm GA, the objective function evaluation incorporates Ghanbari et al. comparison formulas presented in section 2.3 for direct comparison of fuzzy numbers.

The empirical results presented in Table 1 provide compelling evidence of the proposed ABS-GA algorithm's superiority. A critical aspect of this comparison is the allocated computational budget: the standard Genetic Algorithm (GA) was permitted a runtime that is, on average, 20% longer than that of ABS-GA. Despite this deliberate advantage given to the conventional method, ABS-GA consistently produced solutions with significantly superior objective function values across all tested instances. For example, in the largest problem instance (1950 × 2000), ABS-GA achieved a solution that is orders of magnitude better, as reflected by the performance ratio of 78,353.29.

This consistent and substantial advantage leads to a key insight: the performance gain is not merely a result of faster hardware or more computational cycles, but a direct consequence of a fundamental improvement in the problem's mathematical formulation. By integrating the ABS algorithm as a pre-processing step, we effectively transform a high-dimensional constrained problem into a lower-dimensional unconstrained one within the null space of A. This transformation serves two critical purposes. First, it eliminates the curse of dimensionality by reducing the search space from $\mathbb{R}^n$ to $\mathbb{R}^{n-m}$. When

https://doi.org/10.57647/fomj.2026.0701.04

Table 1. Comparison of GA vs. proposed Algorithm (Algorithm 3).

Problem Size	r (Algorithm 3, timed GA)
90 × 100	22260.53
140 × 150	20377.92
190 × 200	30311.48
230 × 250	64954.79
290 × 300	46627.651
350 × 400	83012.01
450 × 500	59155.49
650 × 700	71888.19
750 × 800	100337.98
950 × 1000	78414.62
1150 × 1200	63038.34
1200 × 1300	128684.44
1450 × 1500	59413.95
1650 × 1700	63461.50
1950 × 2000	78353.29

m is close to n, as in our test cases, this reduction is exponential in terms of search volume. Second, it guarantees intrinsic feasibility, rendering the linear equality constraints $Ax = b$ implicit. Consequently, the GA is liberated from the burden of exploring infeasible regions or relying on penalty functions that can distort the fitness landscape. The results in Table 1, therefore, validate that the ABS-GA's power stems from this strategic pre-processing, which allows the GA to focus its entire computational effort on optimizing within a compact, well-defined, and feasible subspace.

Note: To further demonstrate the effectiveness of our proposed ABS-GA algorithm, we conducted additional experiments using standard metaheuristic algorithms such as GA and particle swarm optimization (PSO) implemented in MATLAB on an unconstrained reformulation of the original problem. In this approach, we eliminated the equality constraints $Ax = b$ by incorporating them into the objective function as a quadratic penalty term, transforming the constrained fuzzy optimization problem into an unconstrained one. The penalty parameter was set to a large value 10^6 to strongly penalize constraint violations, and the feasibility tolerance for accepting a solution was set to 10^{-4}. Despite these efforts, both the standard GA and PSO failed to find a single feasible solution across multiple independent runs. Even with a population size of 200 and 500 generations, all generated solutions violated the equality constraints by magnitudes significantly exceeding the tolerance. This failure can be attributed to the fact that the feasible region defined by $Ax = b$ and $x \geq 0$ is a low-dimensional manifold within the high-dimensional space $\mathbb{R}^n$. Without any mechanism to guide the search toward this manifold, the probability of randomly generating points that satisfy the constraints is extremely low. In contrast, our proposed ABS-GA algorithm successfully finds feasible, high-quality solutions for all test instances. This success stems from two key innovations introduced in this paper. First, the ABS algorithm analytically projects the problem onto the null space of matrix A, reducing the dimensionality from n to $n-m$ and ensuring that all solutions generated during optimization automatically satisfy the equality constraints. Second, the novel mutation operator based on tangent cone approximation enables intelligent exploration of the feasible region by generating directions derived from the polar cone of ε-active constraints. This operator, combined with the cone-based initialization strategy described in Algorithm 2, ensures that the population maintains diversity while remaining strictly within the feasible region. Together, these contributions transform a constrained problem that is practically unsolvable by standard metaheuristic methods into a tractable formulation where feasible solutions can be efficiently discovered and optimized.

5. Conclusions and future works

In this paper, we have introduced a novel ABS-GA algorithm for solving fuzzy optimization problems. The key contributions of our approach can be summarized as: First, we employed the ABS algorithm to effectively reduce the problem dimension, thereby decreasing computational complexity and improving solution efficiency. Second, we developed a GA enhanced with a novel mutation operator based on tangent cone approximation, which enables efficient exploration of the feasible region by generating new points from existing solutions. Third, we integrated Ghanbari et al. recently introduced comparison formula for direct comparison of LR fuzzy numbers, eliminating the need for conversion to crisp values and preserving fuzzy information throughout the optimization process. The numerical analysis conducted in this study demonstrates that our proposed ABS-GA algorithm outperforms existing methods in terms of both solution quality and computational efficiency.

Future work will proceed along several directions. We first plan to extend the proposed algorithm to fuzzy quadratic programming problems using a ranking function framework, as suggested in [40]. In addition, due to the general structure of the constraints $Ax = b, x \geq 0$, our method can be naturally adapted to various quadratic optimization problems involving uncertainty. Further efforts will focus on generalizing the approach to other types of fuzzy sets and developing adaptive mechanisms for tuning algorithmic parameters to enhance efficiency and convergence.

Despite the promising performance of the proposed ABS-GA algorithm, an important limitation should be acknowledged: the dimensionality reduction achieved by the ABS algorithm depends critically on the difference $n-m$. In problems where m and n are far apart i.e., when the number of constraints m is much smaller than the number of variables n the reduced dimension $n-m$ remains large, limiting the effectiveness of the dimensionality reduction and potentially diminishing the computational advantages of the proposed approach. Addressing this limitation presents a valuable direction for future research.

Acknowledgment
The first and second authors thank the Research Council of Ferdowsi University of Mashhad and Optimization Laboratory of Ferdowsi University of Mashhad and the third authors thanks the Mosaheb Institute of Mathematics, Kharazmi University for supporting this work.

Authors contributions
Ali Mehrabian: Conceptualization, Methodology, Writing-Original Draft, Supervision, Project Administration. Reza Ghanbari: Formal Analysis, Data Curation, Software, Validation. Khatere Ghorbani-Moghadam: Investigation, Data Collection, Writing-Review & Editing.

Availability of data and materials
The authors declare that the data supporting the findings of this study are available within the paper.

Conflict of interests
The authors assert that they do not have any identifiable conflicting financial interests or personal relationships that might be perceived to influence the work presented in this paper.

References

1. Rahdar S, Ghanbari R, and Ghorbani-Moghadam K. A novel approach for comparing LR fuzzy numbers using a numerical technique. Soft Computing 2026
2. Bhaumika A, Kumar-Roya S, and Li DF. Analysis of triangular intuitionistic fuzzy matrix games using robust ranking. IEEE Transactions on Systems 2017; 33:327–36. DOI: 10.3233/JIFS-161631
3. Bortolon G and Degani R. A review of some methods for ranking fuzzy subsets. Fuzzy Sets and Systems 1885; 15:1–19. DOI: 10.1016/0165-0114(85)90012-0
4. Klir GJ and Yuan B. Fuzzy Sets and Fuzzy Logic: Theory and Application. Prentice Hall 1995
5. Wang X and Kerre EE. Reasonable properties for the ordering of fuzzy quantities (I). Fuzzy Sets and Systems 2001; 118:375–85. DOI: 10.1016/S0165-0114(99)00062-7
6. Buckely JJ and Jowers LJ. Monte Carlo Method in Fuzzy Optimization. Fuzziness and Soft Computing 2008; 222
7. Kerre EE. The use fuzzy theory in electrocardiological diagostics. Approximate Reasoning in decision Analysis 1982 :277–82
8. Wang X and Kerre EE. Reasonable properties for the ordering of fuzzy quantities(II). Fuzzy Sets and Systems 2001; 118:387–405. DOI: 10.1016/S0165-0114(99)00063-9
9. Ghanbari R, Ghorbani-Moghadam K, and Mahdavi-Amiri N. A variable neighborhood search algorithm for solving fuzzy number linear programming problems using modified Kerre's method. IEEE Transactions on Fuzzy Systems 2019; 27:1286–94. DOI: 10.1109/TFUZZ.2018.2876690
10. Dutta P and Limboo B. Bell-shaped fuzzy soft sets and their application in medical diagnosis. Fuzzy information and engineering 2017; 9:67–91. DOI: 10.1016/j.fiae.2017.03.004
11. Zimmermann HJ. Fuzzy set theory and its applications. Dordrecht 2001
12. Maturo F and Fortuna F. Bell-Shaped Fuzzy Numbers Associated with the Normal Curve. Springer 2016 :131–44. DOI: 10.1007/978-3-319-44093-4_13
13. Manickam and Devarasan E. An application of bell shape fuzzy number for fingerprint authentication. Fuzzy information and engineering 2016; 11:1
14. Ghanbari R, Ghorbani-Moghadam K, and Mahdavi-Amiri N. Fuzzy linear programming problems: models and solutions. Soft Computing 2020; 24:10043–73. DOI: 10.1007/s00500-019-04519-w
15. Bellman RE and Zadeh LA. Decision making in a fuzzy environment. Management Science 1970; 17:141–64. DOI: 10.1287/mnsc.17.4.B141
16. Ribeiro RA and Pires FM. Fuzzy linear programming via simulated anneal ing. Kybernetika 1999; 35:57–67
17. Eglese RW. Simulated annealing: A tool for operational research. European Journal of Operational Research 1990; 46:271–81. DOI: 10.1016/0377-2217(90)90001-R
18. Kirkpatrick S, Gelatt CD, and Vecchi MP. Optimization by Simulated Annealing. Science, New Series 1984; 220:671–80. DOI: 10.1126/science.220.4598.671
19. Lin FT. A genetic algorithm for linear programming with fuzzy constraints. Journal of Information Science and Engineering 2008; 24:801–17. DOI: 10.6688/JISE.2008.24.3.9
20. Al-Qudaimi and Kumar A. Comment on "Least-squares approach to regression modeling in full interval-valued fuzzy environment". Soft Computing 2019; 23:10019–27. DOI: 10.1007/s00500-018-3556-4
21. Rabiei MR, Arghami NR, Taheri SM, and Gildeh BS. Least-squares approach to regression modeling in full interval-valued fuzzy environment. Soft Computing 2018; 18:2043–59. DOI: 10.1007/s00500-013-1209-7
22. Chen TY. Pythagorean fuzzy linear programming technique for multidimensional analysis of preference using a squared-distance-based approach for multiple criteria decision analysis. Expert Systems With Applications 2021; 164:113908. DOI: 10.1016/j.eswa.2020.113908

23. Fathy E. A new method for solving the linear programming problem in an interval-valued intuitionistic fuzzy environment. Alexandria Engineering Journal 2022; 61:10419–32. DOI: 10.1016/j.aej.2022.03.047
24. Figueroa-García JC, Hernández G, and Franco C. A review on history, trends and perspectives of fuzzy linear programming. Operations Research Perspectives 2022; 9:100247. DOI: 10.1016/j.orp.2022.100247
25. Jana, Pal M, Emilia-Balas V, and Yager RR. Linear programming problem in a picture fuzzy environment. Advanced Studies in Complex Systems 2024 :197–220. DOI: 10.1016/B978-0-44-322024-1.00013-3
26. Fathy E. A new method for solving the linear programming problem in an interval-valued intuitionistic fuzzy environment. Alexandria Engineering Journal 2022; 61:10419–32. DOI: 10.1016/j.aej.2022.03.047
27. Ebrahimnejad A, Verdegay JL, and Garg H. Signed distance ranking based approach for solving bounded interval-valued fuzzy numbers linear programming problems. International Journal of Intelligent Systems 2019; 34:2055–76. DOI: 10.1002/int.22130
28. Liu Z, Kong M, and Yan L. Novel transformation methods among intuitionistic fuzzy models for mixed intuitionistic fuzzy decision making problems. IEEE Access 2020; 8:100596–607. DOI: 10.1109/ACCESS.2020.2997861
29. Ghorbani-Moghadam K, Ghanbari R, and Salari M. Solving fuzzy LR interval linear systems using nonlinear programming. Fuzzy optimization and modeling journal 2024; 4:33–43. DOI: 10.71808/fomj.2024.1190245
30. Ghosh S and Komar-Roy S. Fuzzy bi-level linear programming problem using TOPSIS approach. Fuzzy optimization and modeling journal 2024; 1:1–10
31. Mahmoodirad A. A novel approach for solving linear programming problems with intuitionistic fuzzy numbers. Fuzzy optimization and modeling journal 2023; 4:1–15. DOI: 10.30495/fomj.2023.1995970.1113
32. Nguyen HT and Walker EA. A First Course in Fuzzy Logic. Chapman & Hall 2000
33. Iris and Cevikcan E. A fuzzy linear programming approach for aggregate production planning. Supply chain management under fuzziness 2014
34. Peidro, Mula J, and Poler R. Fuzzy linear programming for supply chain optimization under uncertainty. International journal of information technology and decision making 2010; 9:373–92. DOI: 10.1142/S0219622010003865
35. Gupta P, Mehlawat MK, Inuiguchi M, and Chandra S. Fuzzy portfolio optimization advances in hybrid multi-criteria methodologies. Springer 2014. DOI: 10.1007/978-3-642-54652-5
36. Sadeghi M and Hosseini HM. Energy supply planning in Iran by using fuzzy linear programming approach (regarding uncertainties of investment costs). Energy policy 2006; 34:993–1003. DOI: 10.1016/j.enpol.2004.09.005
37. Kolda TG, Lewis RM, and Torczon V. Optimization by direct search: New perspectives on some classical and modern methods. SIAM review 2003; 45(3):385–482. DOI: 10.1137/S003614450242889
38. Lewis RM and Torczon V. Pattern search methods for linearly constrained minimization. SIAM Journal on Optimization 2000; 10(3):917–41. DOI: 10.1137/S1052623497331373
39. Sastry K, Goldberg DE, and Kendall G. Genetic algorithms. Springer 2013 :93–117. DOI: 10.1007/978-1-4614-6940-7_4
40. Ghanbari R, Ghorbani-Moghadam K, and Mahdavi-Amiri N. A variables neighborhood search algorithm for solving fuzzy quadratic programming problems using modified Kerre's method. Soft Computing 2019; 23:12305–15. DOI: 10.1007/s00500-019-03966-0
41. Abaffy J, Broyden CG, and Spedicato E. A class of direct methods for linear 375 systems. Numerische Mathematik 1984; 45:361–76. DOI: 10.1007/BF01389710